A Multi-Discipline, Multi-Genre Digital Library for Research and Education

Michael L. Nelson
NASA Langley Research Center, MS 158, Hampton, VA 23681, USA
m.l.nelson@larc.nasa.gov

Kurt Maly, Stewart N. T. Shen
Old Dominion University, Computer Science Department, Norfolk, VA 23529, USA
{maly, shen}@cs.odu.edu

Abstract: We describe NCSTRL+, a unified, canonical digital library for educational and scientific and technical information (STI). NCSTRL+ is based on the Networked Computer Science Technical Report Library (NCSTRL), a World Wide Web (WWW) accessible digital library (DL) that provides access to over 100 university departments and laboratories. NCSTRL+ implements two new technologies: cluster functionality and publishing "buckets". We have extended the Dienst protocol, the protocol underlying NCSTRL, to provide the ability to "cluster" independent collections into a logically centralized digital library based upon subject category classification, type of organization, and genres of material. The concept of "buckets" provides a mechanism for publishing and managing logically linked entities with multiple data formats. The NCSTRL+ prototype DL contains the holdings of NCSTRL and the NASA Technical Report Server (NTRS). The prototype demonstrates the feasibility of publishing into a multi-cluster DL, searching across clusters, and storing and presenting buckets of information.

1. Introduction

Digital libraries (DLs) are an important research topic in many educational and scientific communities and have already become integral part of education at all levels. We have learned of many instances where undergraduate and graduate engineering courses are now supplemented with NASA research results that were previously difficult to obtain. DLs have allowed resource limited colleges and universities access to a wealth of research in a variety of disciplines. However access to these DLs is not as easy users would like. Digital library projects are partitioned by both the discipline they serve (computer science, aeronautics, physics, etc.) and by the format of their holdings (technical reports, video, software, etc.). A recent survey found over 10 existing or recent different World Wide Web (WWW) oriented digital library projects spanning over 5 different disciplines [Esler & Nelson 1998]. In short, each educational and scientific community is hand crafting their own digital library infrastructure.

There are two significant problems with current digital libraries. First, interdisciplinary research and education are difficult because the collective knowledge of each discipline is stored in incompatible DLs that are known only to the specialists in the subject. The second significant problem is that although technical information as well as educational materials created consists of manuscripts, software, datasets, etc., the manuscript receives the majority of attention, and the other components are often discarded [Sobieszczanski -Sobieski 1994].

Old Dominion University and NASA Langley Research Center have established NCSTRL+ to address both of these problems. NCSTRL+ is based on the Networked Computer Science Technical Report Library (NCSTRL) [Davis et al. 1995], which is a highly successful digital library offering access to over 100 university departments and laboratory since 1994, and is implemented using the Dienst protocol [Lagoze et al. 1995]. At the development stage, NCSTRL+ will initially include selected holdings from the NASA Technical Report Server (NTRS) [Nelson et al. 1995] and NCSTRL, providing *clusters* of collections along the dimension of disciplines such as aeronautics, space science, mathematics, computer science, and physics, as well as clusters along the dimension of publishing organization and genre, such as

project reports, journal articles, theses, etc. NCSTRL+ holdings will be published in *buckets* [Nelson et al. 1997], an object-oriented construct for creating and managing collections of logically related information units as a single object. A bucket can contain both different data syntax (PostScript, PDF, Word, etc.) and different data semantics (manuscripts, data files, images, software, etc.)

2. Background

NCSTRL+ has a long lineage. In 1992, the ARPA-funded CS-TR project began [Kahn 1995] as did the Langley Technical Report Server (LTRS) [Nelson et al. 1994]. In 1993, the Wide Area Technical Report Server (WATERS) [Maly et al. 1994] shared a code base with LTRS. In 1994, LTRS launched the NTRS, and the CS-TR and WATERS projects formed the basis for the current NCSTRL. In 1997, NTRS and NCSTRL formed the basis for NCSTRL+. We chose to implement NCSTRL+ using Dienst instead of other digital library protocols such as TRSkit [Nelson & Esler, 1997] because of Dienst's success in several years of production in NCSTRL. Dienst appears to be the most scalable, flexible, and extensible of digital library systems we surveyed [Esler & Nelson 1998]. Dienst also serves as the basis for other digital library projects, including: the Electronic Thesis and Dissertation Project [Fox et al. 1996], the University of Virginia undergraduate engineering thesis project [UVa SEAS 1997] and the ACM SIGIR conference proceedings project (which requires ACM authentication) [ACM 1997].

Our buckets are similar in concept to the "digital objects" first proposed in [Kahn & Wilensky 1995]. It is important to note that many services have had "proto-buckets" in operation for some time. However, they provide only different formats of a single manuscript, or may support the concept of separate pages within a manuscript. They do not support an interface to a collection of related objects such as the manuscript, software, datasets, etc. We chose the term "buckets" because related terms such as "objects", "packages" and "containers" are greatly overloaded in the computer science realm and because "buckets" provide a clear visual metaphor for the concept when speaking with non-computer scientists.

3. Clusters of Dienst Servers

Clusters are a way of aggregating logically grouped sub-collections in a DL along some criteria. NCSTRL+ provides 3 clusters: organization, data genre, and subject category (see [Fig. 3] for an example). *Genre* is a term provided by E. Fox in a private communication and refers to distinguishing between journal articles, technical reports, theses and dissertations, etc. For the purposes of this paper, we illustrate the concept of clusters by discussing the subject category cluster. Other clusters are implemented similarly.

Dienst currently carries no concept of subject category in its protocol, despite having provisions for specifying keywords from the title, authors, and abstract. In fact, digital libraries using the Dienst protocol such as NCSTRL have the implicit assumption that all holdings are computer science related. We propose to modify Dienst by providing *cluster* arguments to existing message verbs. We have used a set of message verb modifications to demonstrate the concept of subject category based server cluster functionality. The new clustering service will solve the general case of the problem, where our Dienst modifications will support the specific clustering around subject categories in the early stages of the NCSTRL+ prototype. The purpose of our cluster prototype is to perform experiments with an initial set of clusters and determine user response. NCSTRL+ reads its known subject categories from a preference file thus the list of subjects can be easily replaced or augmented.

4. Buckets

Buckets are a construct for creating publishing and archival entities for digital libraries. A bucket corresponds to a single logical collection of information. Buckets are designed to be highly customizable and unique. Bucket architecture is illustrated in [Fig. 1]. Large archives could have buckets with many different functionalities. Not all bucket types or applications are known at this time. However, we can describe a generalized bucket as containing many formats of the same data item (PS, Word, Framemaker,

etc.) but more importantly, it can also contain collections of related non-traditional STI materials (manuscripts, software, datasets, etc.) Thus, buckets allow the digital library to address the long standing problem of ignoring software and other supportive material in favor of archiving only the manuscript [Sobieszczanski -Sobieski 1994] by providing a common mechanism to keep related STI products together. A single bucket can have multiple *packages*. Packages can correspond to the semantics of the information (manuscript, software, etc.), or can be more abstract entities such as the metadata for the entire bucket, bucket terms and conditions, pointers to other buckets or packages, etc. A single package can have several *elements*, which are typically different file formats of the same information, such as the manuscript package having both PostScript and PDF elements.

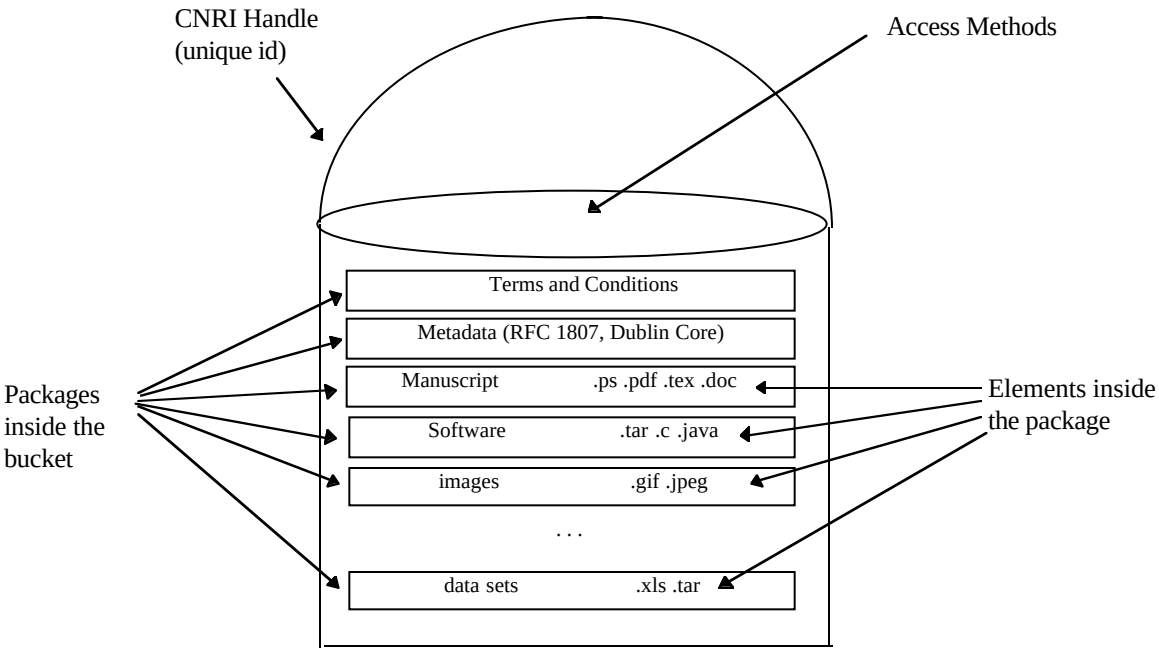

Figure 1: Bucket Architecture

4.1 Bucket Requirements

Buckets are intended to be either standalone objects or to be placed in digital libraries. They have unique ids (CNRI handles) associated with them. Buckets are intended to be useful even in repositories that are not knowledgeable about buckets in general, or possibly just not about the specific form of buckets. Buckets should not lose functionality when removed from their repository. The envisioned scenario is that NCSTRL+ will eventually have moderate numbers of (10s - 100s of thousands) of intelligent, custom buckets instead of large numbers (millions) of homogenous buckets. With buckets, the repository intelligence and functionality can be split between the repository and individual buckets. This could be most useful when individual buckets require custom terms and conditions for access (security, payment, etc.). A bucket gains some repository intelligence as it is extracted from the archive en route to becoming a standalone bucket. A full discussion of bucket functionality can be found in [Nelson et al 1997].

4.2 Bucket Tools

There are two main tools for bucket use. One is the *author tool* [Fig. 2], which allows the author to construct a bucket with no programming knowledge. Here, the author specifies the metadata for the entire bucket, adds packages to bucket, adds elements to the packages, provides metadata for the packages, and selects applicable clusters (which lead to the cluster options available as shown in [Fig. 3]. The author tool gathers the various packages into a single component and parses the packages based on rules defined at the

author's site. Many of the options of the author tool will be set locally via the second bucket tool, the *management tool*. The management tool provides an interface to allow site managers to configure the default settings for all authors at that site. The management tool also provides an interface to query and update buckets at a given repository. Additional methods can be added to buckets residing in a repository by invoking the `add_method` on them and transmitting the new code. From this interface, the manager can halt the archive and perform operations on it, including updating or adding packages to individual buckets, updating or adding methods to groups of buckets, and performing other archival management functions.

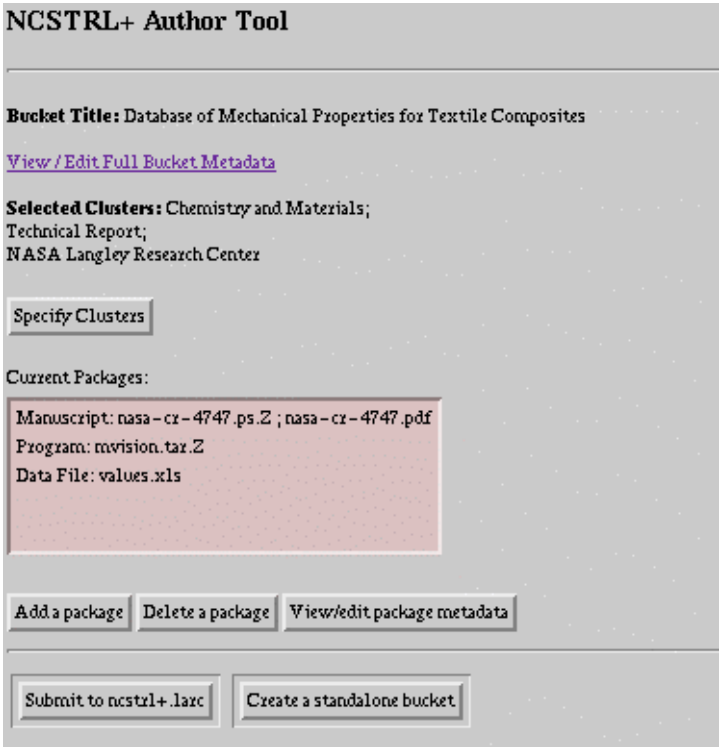

Figure 2: Author Tool

5. Using NCSTRL+

5.1 Searching NCSTRL+

NCSTRL+ searching is similar to searching NCSTRL, with the addition of specifying desired clusters to search. How the advanced fielded search form of NCSTRL+ is modified, allowing the selection of desired subject categories and data genres, is shown in [Fig. 3]. A search results page includes the keyword and cluster hit results. The user will select the desired bucket from this page. At that point, the bucket will return the defined default initial interface of the bucket, which will be dependent on the bucket contents and the rules present. In practice, the bucket presentation will look largely similar to the choices available to current users of NCSTRL. This is especially true if the buckets in which they are interested only contain various manuscript formats. However, the real benefit is the richer presentation formats available if the bucket has non-manuscript packages. The bucket interface is similar to NCSTRL, with the exception that the additional data semantics are presented (software, datasets, etc.).

5.2 Publishing into NCSTRL+

The goal of NCSTRL+ is to produce the least intrusive interface possible to the author. The authoring process for NCSTRL+ is to be as similar to authoring into NCSTRL as possible. Additions include the ability to add to a bucket multiple data semantics and formats through using multiple selection

boxes to select local files. Publishing a manuscript in NCSTRL is equivalent to publishing a package in NCSTRL+, and publishing a bucket is the sum of publishing all of its packages. The author also has to choose the appropriate cluster to place the new bucket in. This step can be skipped if the site manager has defined a default, or if authors have saved a value already in their preferences.

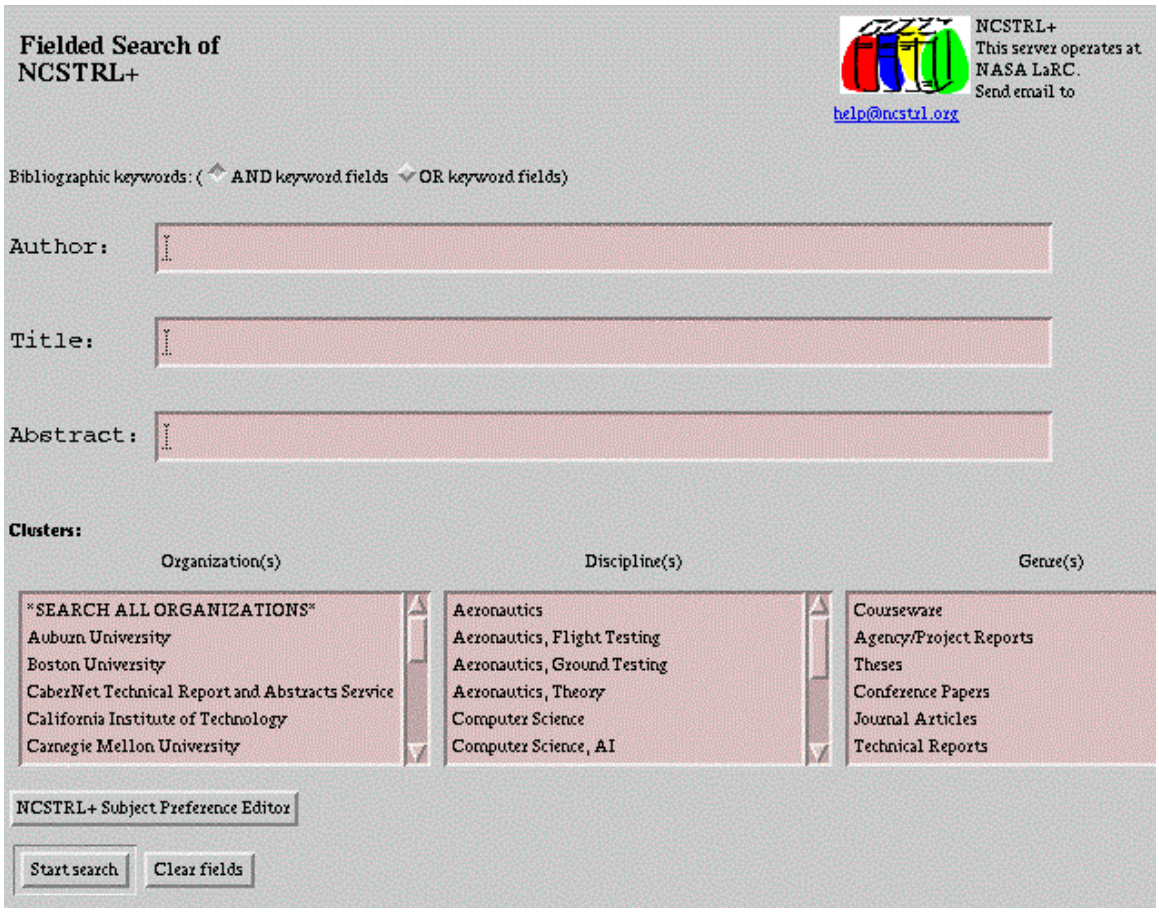

Figure 3: The Fielded Search Screen of NCSTRL+

6. Status and Future Work

We are using the author tool to populate NCSTRL+ to gain insight on how to improve its operation. We are starting with buckets authored at Old Dominion University and NASA Langley Research Center and are choosing the initial entries to be "full" buckets, with special emphasis on buckets relating to NSF projects for ODU and for windtunnel and other experimental data for NASA. Until NCSTRL+ becomes a full production system, we are primarily seeking rich functionality buckets that contain diverse sets of packages.

It is also important to note that adding a subject category mechanism to NCSTRL+ provides the necessary groundwork for additional services for digital libraries using Dienst. These could include subject-based browsing of NCSTRL+ holdings, as well as selected dissemination of information (SDI). This would be most useful if users were offered a subscription option to receive digested updates (i.e., e-mail messages) of new additions to NCSTRL+ in specified subject areas. The initial defined subject categories for NCSTRL+ and cross-listing them with other subject-specific categorization schemes is intended to provide a working framework for evaluating the prototype. As more experience in NCSTRL+'s use is gained, the fine tuning of the subject categories and appropriate cross listing becomes an area that would benefit from the attention of a professional cataloger.

7. Conclusions

To meet the increased requirements for multidisciplinary activities in educational and scientific communities, we have prototypes of NCSTRL+ and are in the process of full implementation. The most significant technology from this project is the concept of buckets as a construct to capture multiple data formats and genres in an intuitive manner. NCSTRL+ provides a platform for experimentation for testing user response to multidiscipline clusters and logical collections of STI. We are in the process of experimenting with users at NASA and Old Dominion University. From the users' perspective, the publishing and searching interfaces are largely unchanged. However, it is unknown what impact the cluster and bucket modifications have on network load, search and retrieval times, the users' perceived quality of searching multiple clusters, etc. To determine these unknowns, NCSTRL+ will have to grow to a large enough size to be considered a useful production system. The authors seek other users and participants for NCSTRL+.

8. References

[ACM 1997]. ACM SIGIR On-Line Conference Proceedings (1997), http://turing.acm.org:8071/

[Davis et al. 1995]. Davis, J., Krafft, D., & Lagoze, C. (1995).Dienst: Building a Production Technical Report Server, *Advances in Digital Libraries*, Springer-Verlag, 211-222.

[Esler & Nelson 1998]. Esler, S. & Nelson, M. (1998). The Evolution of Scientific and Technical Information Distribution, *Journal of the American Society of Information Science*, 49(1), 82-91.

[Fox et al. 1996]. Fox, E., Eaton, J., McMillan, G., Kipp, N., Weiss, L., Arce, E., & Guyer, S. (1996), National Digital Library of Theses and Dissertations: A Scalable and Sustainable Approach to Unlock University Resources, *D-Lib Magazine*, http://www.dlib.org/dlib/september96/theses/09fox.html

[Kahn 1995]. Kahn, R. (1995). An Introduction to the CS-TR Project, http://WWW.CNRI.Reston.VA.US/home/describe.html

[Kahn & Wilensky 1995]. Kahn, R. & Wilensky, R. (1995). A Framework for Distributed Digital Object Services, cnri.dlib/tn95-01, http://www.cnri.reston.va.us/home/cstr/arch/k-w.html

[Lagoze et al. 1995]. Lagoze, C., Shaw, E., Davis, J. & Krafft, D, (1995), Dienst: Implementation Reference Manual, Cornell University Computer Science Technical Report TR95-1514.

[Maly et al. 1994]. Maly, K., French, J., Selman, A. & Fox, E, (1994). Wide Area Technical Report Service, *Proceedings of the Second International World Wide Web Conference*, Chicago, IL, 523-533.

[Nelson et al. 1995]. Nelson, M., Gottlich, G., Bianco, D., Paulson, S., Binkley, R., Kellogg, Y., Beaumont, C., Schmunk, R., Kurtz, M., & Accomazzi, A. (1995). The NASA Technical Report Server, *Internet Research: Electronic Networking Applications and Policy*, 5(2), 25-36.

[Nelson et al. 1994]. Nelson, M., Gottlich, G. & Bianco, D. (1994), World Wide Web Implementation of the Langley Technical Report Server, NASA TM-109162.

[Nelson et al. 1997]. Nelson, M., Maly, K. & Shen, S. (1997), Buckets, Clusters and Dienst, Old Dominion University Computer Science Technical Report 97-30. (Also available as NASA TM-11287)

[Nelson & Esler 1997]. Nelson, M. & Esler, S. (1997), TRSkit: A Simple Digital Library Toolkit, *Journal of Internet Cataloging*, 1(2), 41-55.

[Sobieszczanski-Sobieski 1994]. Sobieszczanski-Sobieski, J. (1994). A Proposal: How to Improve NASA-Developed Computer Programs, NASA CP-10159, 58-61.

[UVa SEAS 1997]. UVa SEAS Electronic Undergraduate Thesis Pilot (1997). http://univac.cs.virginia.edu:3066/SEAS_ETD.html